Smartphone-based Indoor Localization for Blind Navigation across Building Complexes

Masayuki Murata*, Dragan Ahmetovic†, Daisuke Sato*, Hironobu Takagi*, Kris M. Kitani†, Chieko Asakawa†‡
*IBM Research - Tokyo, Japan, †Robotics Institute, Carnegie Mellon University, USA, ‡IBM Research, USA
Email: muratams@jp.ibm.com, dragan1@cmu.edu, {dsato, takagih}@jp.ibm.com, {kkitani, chiekoa}@cs.cmu.edu

Abstract—**Continuous and accurate smartphone-based localization is a promising technology for supporting independent mobility of people with visual impairments. However, despite extensive research on indoor localization techniques, they are still not ready for deployment in large and complex environments, like shopping malls and hospitals, where navigation assistance is needed. To achieve accurate, continuous, and real-time localization with smartphones in such environments, we present a series of key techniques enhancing a probabilistic localization algorithm. The algorithm is designed for smartphones and employs inertial sensors on a mobile device and Received Signal Strength (RSS) from Bluetooth Low Energy (BLE) beacons. We evaluate the proposed system in a 21,000 m^2 shopping mall which includes three multi-story buildings and a large open underground passageway. Experiments in this space validate the effect of the proposed technologies to improve localization accuracy. Field experiments with visually impaired participants confirm the practical performance of the proposed system in realistic use cases.**

I. Introduction

To enable automated smartphone-based turn-by-turn navigation assistance for people with visual impairments in complex large-scale real-world scenarios, it is important to improve smartphone-based localization accuracy over existing technologies, *e.g.*, GPS, WiFi, BLE beacons. Since many of the exiting technologies have been evaluated only in controlled environments, it is not clear whether current localization algorithms for mobile navigation can be successful in building-scale real-world scenarios, such as complex public facilities and commercial buildings.

When we consider localization systems for turn-by-turn navigation in real world scenarios for people with visual impairments, the requirements are quite challenging. We identify four key challenges, that are often overlooked in the research literature.

Accurate and Continuous Localization. It is critical to achieve both *accurate* and *continuous* localization when giving turn-by-turn instructions to people with visual impairments. Based on preliminary tests, a localization accuracy of less than 2 meters is desirable to provide timely turn-by-turn guidance, especially at decision points such as corridor intersections or entrances. A higher localization error could lead a person with visual impairment through the wrong door or cause collisions with the environment. While probabilistic localization algorithms are designed to deal with a certain level of noise, many approaches can fail catastrophically when the fidelity of the current state estimate degrades. Remedies to such failures (*e.g.*, modifications to the state sampling process [1], [2]) have been proposed, however, such approaches can cause the location estimates to jump around discontinuously. There is no tolerance for such instability when guiding people with visual impairments.

Scaling to Multi-Story Buildings. Previous methods for localizing users with a smartphone have primarily focused on 2D floor plans [3]. However, most buildings in metropolitan areas are multi-story buildings. In particular, in public facilities such as shopping centers or subway stations, people constantly transition from floor to floor and from building to building. It is therefore critical to localize users across floors and during floor transitions. To the best of our knowledge, work addressing accurate localization over multi-story buildings has been limited.

Signal Bias Adaptation at Scale. Different mobile devices observe different receiver signal strength (RSS) values from the same signal transmitter at the same location due to differences in reception sensitivity of the underlying radio hardware. Previous works have addressed this issue [4], [5], [6] by estimating a signal strength offset value. However in real world applications, the number and strength of observable transmitters, *e.g.,* beacons, changes dynamically over time. There is limited prior work addressing this challenge, *i.e.,* varying RSS values over multiple devices in dynamic situations.

Scaling to Large Numbers of Measurements. As the size of deployment grows to building-scale proportions, the computational costs of regression algorithms for accurately mapping between locations and RSS observations grow prohibitively expensive to run in real-time with smartphone resources. Efficient methods are needed to accelerate the computation of localization for building-scale environments.

Contributions. To deal with the aforementioned challenges within a unified framework, we use a probabilistic localization algorithm as our foundation and enhance it with a series of novel innovations:

1) *Localization Integrity Monitoring* introduces an internal state machine to allow the system to be softly reinitialized during failures,
2) *Floor Transition Control* uses changes in barometric pressure and RSS values to regularize localization during elevator or escalator use,
3) *Adaptive Signal Calibration* uses a Kalman filter to incrementally estimate signal offsets,

4) *Fast Likelihood Computation* is realized through the use of a set of locally trained truncated regression models.

We implement the localization system and perform a thorough evaluation with data collected in a large, complex indoor environment composed of three multi-story buildings and a broad underground pedestrian walkway. To quantitatively validate the reliability of our method, we collected ground truth localization data using a Light Detection and Ranging (LIDAR) sensor. On the basis of this data, we evaluate the effect of our proposed enhancement modules. We also evaluate the localization accuracy in experiments with visually impaired participants. The localization system is integrated into a turn-by-turn navigation application on iOS devices, and we evaluate localization error while the participants are traversing the environment with the aid of the navigation app.

II. Related work

A. Indoor Localization

Indoor localization has been extensively studied for the past two decades [7], [8], [9], [10], [11]. Among various indoor localization techniques, localization based on RSS of wireless signals such as WiFi or Bluetooth is one of the most popular [7], [12], [1], [13], [14] due to its use of off-the-shelf mobile devices, potential for high accuracy, and relatively low infrastructure cost.

Besides RSS-based methods, various localization techniques have been developed based on RFID [15], UWB radios [16], ultrasound [17], *etc.* Most of these approaches require specialized hardware for either or both of infrastructure and user. Image based localization methods (*e.g.*, [6]) are promising, however they are not robust enough in scenes having few visual features and appearance changes. Recently, Channel State Information (CSI) has been investigated to achieve a higher accuracy ($\sim$1m) localization [18], [19] with WiFi than RSS-based methods. Unfortunately, CSI is not available on commodity smartphones, and therefore it cannot yet be used for our envisioned use case.

RSS-based localization methods can be divided into two categories: fingerprint-based or model-based methods. Fingerprint-based localization is the prevalent solution to achieve and guarantee better accuracy [7], [20], [12], [1], [6]. It is usually conducted in two phases: offline training (site-survey) phase to collect RSS data labelled with correct locations and online operating phase to estimate the location of a user's mobile device. Model-based methods assume a propagation model of radio waves to estimate RSS at various locations. For this purpose, the log-distance path loss model is widely used [21], [22]. Although model-based methods require much less training data than fingerprint-based methods, they are also less accurate, particularly in non-line-of-sight conditions. To reduce site-survey effort, localization methods relying on fingerprint database construction by unsupervised learning or crowdsourcing have recently been proposed [22], [23]. However, they are less accurate than fingerprint-based methods and insufficient for applications with high accuracy requirements.

WiFi fingerprinting is widely studied for indoor localization using RSS of WiFi signals, since access points (APs) are ubiquitous in most environments [7], [12], [1], [11]. However, WiFi coverage is not tuned to provide accurate localization, and WiFi AP positioning depends on environment wiring constraints and connectivity requirements.

In contrast, BLE based localization has gained prominence following the introduction of the BLE protocol standard and the commercialization of off-the-shelf BLE beacon transmitters [24], [14], [25]. Compared to WiFi APs, BLE beacons are small, low-cost (\$5–20 per device), and have low power consumption. Therefore, they can be battery powered and placed with fewer constraints than WiFi APs, thus achieving uniform and controlled coverage, which results in consistent and higher levels of localization accuracy. Also, RSS from BLE beacons are accessible on most commodity smartphone operating systems (iOS and Android) whereas WiFi scanning is currently prohibited on iOS devices.

To further improve the localization accuracy achieved by RSS fingerprint-based methods, recent approaches also perform fusion between RSS-based localization, and user's motion model [10]. The fusion algorithms are used to integrate the movement of a user obtained with pedestrian dead reckoning (PDR) methods, applied on data from sensors embedded on a mobile device [26], [27], and RSS fingerprint-based methods. This way, it is possible to produce more accurate localization from noisy observations. The particle filter is one of the most successful sensor fusion algorithms for localization [12], [1], due to its excellent extendability and ease of implementation. This approach is also suitable for indoor navigation because it is based on state space modeling and it is possible to estimate unobservable variables, *e.g.*, user's walking direction, which are important for navigation. Because of these advantages, we adopt this approach as our base localization framework. Details of our implementation of the particle filter algorithm are presented in Section III.

B. Navigation Assistance for People with Visual Impairments

Prior research investigated various indoor navigation approaches to support people with visual impairments [28], [29]. Among these, smartphone-based turn-by-turn navigation systems using BLE beacons provide accurate navigation assistance, use off-the shelf infrastructure, and they are easy to deploy [30], [31], [32], [33], [34]. Turn-by-turn navigation is a guidance method that orients a user towards a destination through sequential vocal messages. These messages can be announcements of distances, action instructions and additional contextual information (*e.g.*, nearby points of interest). Of these, action instructions are the most important since they notify the user of the actions to perform, for example, "Turn right at the corner" or "Proceed 10 meters after opening the door".

From the perspective of localization methods, systems such as *StaNavi* [30] and *GuideBeacon* [31] are based on proximity sensing using the RSS of BLE beacons. *NavCog* [32], [33] adopted accurate localization on a simplified space representation using one-dimensional edges for ease of deployment [35]. Those methods have limited pose estimation capabilities and localization accuracy, which are not suitable for navigation in complex environments. To enable navigation in large-scale environments such as multi-story shopping malls, *NavCog3* [34] is designed to rely on a location tracking method to exploit user's position and heading direction.

Sato et al. [34] focus on the user interface and usability of the navigation assistant. In contrast, our paper demonstrates the localization system design to achieve indoor localization with levels of accuracy that can be applicable for navigating people with visual impairments in building-scale environments.

III. Base Localization Model

We now describe the underlying probabilistic framework for state estimation using the particle filter. We give a sketch of the base particle filter model for localizing a user with smartphone sensors and a BLE beacon network. The basic concepts introduced here will help to contextualize the key technical innovations introduced later in Section IV.

A. Particle Filter

The particle filter is an efficient algorithm for continuously estimating a user's location [2], [12], [1]. Let t be an index of time, $\boldsymbol{z}_t$ be the user's state (*e.g.*, 2D location), $\boldsymbol{r}_t$ be the sensors measurements (*e.g.*, RSS values from multiple transmitters), $\boldsymbol{u}_t$ be the control input (*e.g.*, the user's movement obtained from inertial measurements), and m be the map. The particle filter approximates the posterior of a user's state conditioned on all previous measurements $\boldsymbol{r}_{1:t}$ and inputs $\boldsymbol{u}_{1:t}$ by a finite set of samples (particles) $\{\boldsymbol{z}^l_{t|t}\}^L_{i=1}$, where l and L correspond to the index and the total number of particles.

Three steps are iteratively performed in the algorithm:

1) Predict each particle's state using the motion model, $p\left(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{u}_t, m\right)$ which describes the relationship between the user's previous state $\boldsymbol{z}_{t-1}$ and the user's current state $\boldsymbol{z}_t$ given the control input $\boldsymbol{u}_t$.
2) Compute each particle's importance weights using the observation model, $p\left(\boldsymbol{r}_t|\boldsymbol{z}_t\right)$ which describes the likelihood of the observed signal strength measurements $\boldsymbol{r}_t$ at user's state $\boldsymbol{z}_t$.
3) Resample particles with replacement from a predicted particle set according to the importance weights.

B. Motion Model

The motion model $p\left(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{u}_t, m\right)$ predicts the user's location using sensory information (*e.g.,* sensors on the user's smartphone). We model this distribution using Gaussian random variables where the predicted location of the user (x_t, y_t), under the control input $\boldsymbol{u}_t = (s_t, \theta_t)^T$, can be written as:

$$x_t = x_{t-1} + s_t v_t \cos(\theta_t + \theta^{\circ}_t)\Delta t + \xi_{x,t} \quad (1a)$$
$$y_t = y_{t-1} + s_t v_t \sin(\theta_t + \theta^{\circ}_t)\Delta t + \xi_{y,t}. \quad (1b)$$

The motion state of the user s_t is an indicator function (*i.e.,* user is moving: $s_t = 1$ or stopped: $s_t = 0$), θ_t is the orientation of the smartphone and θ°_t is the offset of the orientation between the user and the smartphone. v_t is the velocity of the user and Δt is the time step between $t-1$ and t. The cosine and sine represent the direction of displacement of the user location for Δt on the x-y plane. The user's motion state s_t can be detected by thresholding the standard deviation of the magnitude of acceleration in a short time window [36]. The attitude of the smartphone can be obtained by integrating smartphone IMU sensor data (*i.e.*, accelerometer and gyroscope data). $\xi_{x,t}$ and $\xi_{y,t}$ correspond to perturbation noise to x_t and y_t with zero mean Gaussian random variables. The θ°_t and v_t are estimated through particle filtering by incorporating them into a state vector as $\boldsymbol{z_t} = [x_t, y_t, v_t, \theta^{\circ}_t]^{\top}$.

C. Observation Model

The observation model $p\left(\boldsymbol{r}_t|\boldsymbol{z}_t\right)$ describes the likelihood of the sensor measurements $\boldsymbol{r}_t$ given the user's state $\boldsymbol{z}_t$. Our method assumes that a dense network of BLE beacons are installed in the environment similar to [14]. The observation model is learned from training data, as a function that maps position to RSS. Formally, we are given a set of training samples $D = \{(\boldsymbol{x}_n, r_n)\}^N_{n=1}$, where $\boldsymbol{x}_n$ is the input (location) and r_n is the output (RSS), and N is the number of training samples. We apply kernel ridge regression [37] to predict RSS given a vector of location $\boldsymbol{x}_*$ as:

$$\mu(\boldsymbol{x}_*) = m(\boldsymbol{x}_*) + k^T_*(K + \sigma^2_n I)^{-1}(\boldsymbol{r} - m(\boldsymbol{X})) \quad (2)$$

where k_* is the vector of the kernel function between $\boldsymbol{x}_*$ and each training point, K is the matrix of the kernel function relating each pair of points, σ_n is a regularization parameter, and $m(\boldsymbol{x}_*)$ is an explicit prior on the mean function. The mean function $m(\boldsymbol{x}_*)$ is computed using the well-known log-distance path loss model as [38]:

$$m_i(\boldsymbol{x}) = -10 n_i \log\left(d(\boldsymbol{x}, b_i)\right) + A_i \quad (3)$$

where $d(\boldsymbol{x}, b_i)$ is the physical distance between position $\boldsymbol{x}$ and BLE beacon b_i, n_i is the decay exponent, and A_i is the path loss at the reference distance of 1 m. The variance of the output, σ^2_i, is separately estimated as a position-independent constant for each BLE beacon b_i. As a result, the RSS for i-th beacon is modeled by

$$p(r_{i,t}|\boldsymbol{x}_t) = N(r_{i,t}; \mu_i(\boldsymbol{x}_t), \sigma^2_i). \quad (4)$$

The likelihood model, given all of the RSS values from multiple beacons, is obtained from the product of single beacon likelihood terms. We use only the top K RSS signals to accelerate computation time. Formally, Let $\boldsymbol{r}_t = (r_{1,t}, \ldots, r_{K,t}, \ldots, r_{M,t})^{\top}$ be the values of an RSS vector in descending order. The observation model with a limited number of beacons can be written as:

$$p(\boldsymbol{r}_t|\boldsymbol{x}_t) = \prod^K_{i=1} p(r_{i,t}|\boldsymbol{x}_t)^{\alpha} \quad (5)$$

where M is the total number of observed beacons, K is the number of beacons used for localization, and α is a smoothing coefficient to prevent the likelihood model from overconfident estimates due to dependencies between $r_{i,t}$ [2].

D. Initial Pose Estimation

When the localization algorithm starts tracking, it must identify the initial pose, including the location and orientation. Specifically, we use the Metropolis-Hastings algorithm [39] to draw samples from $p(\boldsymbol{x}_t|\boldsymbol{r}_t)$ (equivalent to sampling from $p(\boldsymbol{r}_t|\boldsymbol{x}_t)$ when $p(\boldsymbol{x}_t)$ is assumed to be uniform) to estimate the location. To compute the initial orientation we use the smartphone magnetometer and GPS receiver, but place a very large variance on the distribution to account for uncertainty.

IV. Proposed Technical Innovations

The base particle filtering framework described above is sufficient to localize a user in ideal situations. However, in real-world scenarios, the system can fail catastrophically and drain smartphone computing resources if key issues are not taken into consideration. This section introduces four key technical innovations, based on lessons learned from real-world use cases, that allow us to scale smartphone-based localization to very large (building-scale) environments.

A. Localization Integrity Monitoring

The Localization Integrity Monitoring (LIM) module observes whether the localization is working as expected and switches the behavior of the system in times of uncertainty. Specifically, we implement a state machine to monitor the integrity of localization to control how sensor inputs are used by the localization algorithm. Figure 1 shows a diagram of the state machine. The state machine consists of four states: *unknown, locating, tracking,* and *unreliable*. The state starts from *unknown* and changes depending on the RSS vector input $\boldsymbol{r}_t$. After a first RSS vector input is given, the state changes to *locating*, in which the localization system begins initialization. The *locating* state is repeated until the uncertainty of the current location decreases below a certain level. Once initialized the state transits to *tracking*. If the uncertainty remains high, the *locating* state returns to *unknown*. In the *tracking* state, the localization system tracks the user's state via the particle filter.

The important issue here is the reaction of the system when the *tracking* state changes to the *unreliable* state. Formally, let $X_{t|1:t}$ and $X_{t|t}$ be a set of particles approximating the belief distribution $p(\boldsymbol{x}_t|\boldsymbol{r}_{1:t}, \boldsymbol{u}_{1:t})$ and a set of particles drawn from $p(\boldsymbol{x}_t|\boldsymbol{r}_t)$, respectively. The set of particles $X_{t|1:t}$ is obtained as the filtered states by the particle filter and $X_{t|t}$ can be obtained by generating samples using the method in Section III-D. Abnormality is measured as the ratio of the maximum likelihood given $X_{t|1:t}$, and the maximum likelihood given $X_{t|t}$, and it is formally defined as:

$$a(X_{t|1:t}, X_{t|t}, \boldsymbol{r}_t) = \frac{\max\limits_{x_t \in X_{t|1:t}} p(\boldsymbol{r}_t|\boldsymbol{x}_t)}{\max\limits_{x_t \in X_{t|t}} p(\boldsymbol{r}_t|\boldsymbol{x}_t)}. \tag{6}$$

The numerator and denominator take similar values when the device location is successfully tracked. Otherwise, the numerator is much smaller than the denominator because the region with high likelihood is not covered with the tracked

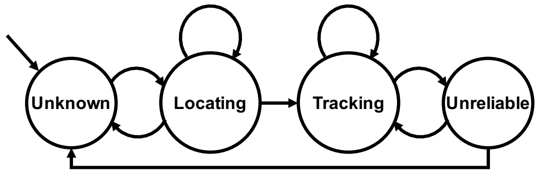


Fig. 1: Localization Integrity Monitoring

particles. As a result, an abnormal situation can be detected by checking whether $a(X_{t|1:t}, X_{t|t}, \boldsymbol{r}_t)$ takes an exceedingly small value, e.g., 0.01. When an abnormal situation is detected, the *locating* state changes to the *unreliable* state. The role of the *unreliable* state is to buffer the decision to revert the state to *unknown*. When an abnormal situation is subsequently detected during the *unreliable* state, the state moves to *unknown*. In this state, the localization system stops to update the unreliable belief distribution conditioned by past inputs and restarts localization from the initialization.

B. Floor Transition Control

The Floor Transition Control (FTC) module seamlessly bridges the estimated location of the user from a source floor to a target floor to reduce the failure in localization related to floor transitions. To apply the localization algorithm in multi-story environments, we need to introduce transitions between two-dimensional floors into the localization algorithm. We denote the floor on which a user is as f_t and augment the location vector $\boldsymbol{x}_t$ as $\boldsymbol{x}_t = (x_t, y_t, f_t)^\top$.

We also define a subset of states called *floor transition areas* which include staircases, escalators and elevators. When the motion model predicts the user's state, floor to floor transitions can only occur at a floor transition area. In cases where the user's location is estimated on a escalator, the motion model adds a constant velocity motion to the user's state to model the passive movement of the user carried by the escalator without taking any steps.

We propose a multi-modal observation model to allow seamless transitions between floors. The standard observation model defined above becomes unstable when transiting from floor to floor because the number of visible beacons signals can be very sparse in that situation (*e.g.*, due to elevator walls blocking BLE signal). To reduce the risk of increased error during floor transition, we use the smartphone barometer in addition to beacon RSS to actively update the user's location. Although it has been shown that barometer readings alone are very noisy [40], they can be used in conjunction with RSS to detect floor changes. The barometer can be used to detect changes in height by thresholding the standard deviation of estimated height over a short period of time.

Formally, let variable c_t denote a detected floor change (*i.e.*, changing: $c_t = 1$ or not changing: $c_t = 0$) and A_T be floor transition areas including stairs, escalators or elevators. We introduce a conditional motion model and a modification to the observation model to take into account changes in floors. Specifically, we introduce $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, c_t)$ and $p(c_t|\boldsymbol{x}_t)$, into the particle filter. In the motion model, the location of a particle $\boldsymbol{x}_{t-1}^l$ is exchanged to the closest point in A_T with a probability, $\max(0, p_{A_T} - \sum_{\boldsymbol{x}_t^l \in A_T} w(\boldsymbol{x}_t^l))$, where p_{A_T} is an acceptable lower limit for $p(\boldsymbol{x}_t \in A_T | c_t = 1)$. In the observation model, detected height changes are used to update the weights of the particles as:

$$w_t(\boldsymbol{x}_t^l) \leftarrow \frac{w_t(\boldsymbol{x}_t^l) p(c_t = 1 | \boldsymbol{x}_t^l)}{\sum_{k=1}^L w_t(\boldsymbol{x}_t^k) p(c_t = 1 | \boldsymbol{x}_t^k)} \tag{7}$$

where $p(c_t = 1 | \boldsymbol{x}_t)$ is the likelihood of $\boldsymbol{x}_t$ given $c_t = 1$ which satisfies that $p(c_t = 1 | \boldsymbol{x}_t \notin A_T) < p(c_t = 1 | \boldsymbol{x}_t \in A_T)$.

In addition to obtaining reliable estimates of location at all times, we also desire to have predictive location estimates (before the user arrives at a location) to allow the navigation application to issue instructions as early as possible. Further enhancements to the floor transition module can be achieved by exploiting the observation model of BLE beacon RSS observations. In addition to the sampling of the floor variable in the motion model, the floor variable is selectively sampled by the observation model when a floor change is detected ($c_t = 1$). More specifically, the floor variable is drawn according to the likelihood of the observation where variables, except for the floor, are fixed.

$$f_t^l \sim p(\boldsymbol{r}_t | f_t^l, \boldsymbol{x}_t^{l,\setminus f}) \tag{8}$$

where $\boldsymbol{x}_t^{l,\setminus f}$ denotes the location of the l-th particle except for the floor variable f_t^l. This sampling scheme improves the response of the estimated location to actual floor transitions. In practice, this allows us to localize the user few seconds before the actual floor arrival and gives us enough time to notify the user.

C. Adaptive Signal Calibration

The Adaptive Signal Calibration (ASC) module adjusts the device RSS offset with a time-series filtering algorithm modified for a truncated observation vector. A basic approach for adjusting signal offset can be implemented by minimizing the distance in RSS space between two devices with respect to a RSS offset [5], [6]. We pose a similar optimization problem to calibrate for offsets in signal reading but apply the updates in an online manner using a lightweight time-series filter based on the Kalman filter [2]. This approach also enables automatic adaptation to the uncertainty in the offset estimation due to dynamical changes in the number of observed beacons.

Formally, we denote RSS $r_{i,t}$ of device i at time step t. To differentiate the RSS observed during training time and test time, we use the notation $r_{i,t}^A$ for the training time signal and $r_{i,t}^B$ as the test time signal. We denote the current predicted mean of RSS $\mu_i(\boldsymbol{x}_t)$ in shorthand notation as $\mu_{i,t}$. We assume that at test time, there will be an offset introduced to the RSS due to a different device being used (*e.g.*, a different iPhone) or some global changes to the signal environment (*e.g.*, weakening of the beacon signal strength). Formally, we assume

$$r_{i,t}^B = r_{i,t}^A + r_t^{\mathrm{o}} \tag{9}$$

where r_t^{o} is the signal strength offset at time t.

Since the observation model $p(r_{i,t}^A | \boldsymbol{x}_t) = N(r_{i,t}^A; \mu_{i,t}, \sigma_{i,t}^2)$ is Gaussian, we can denote the test time observation model as:

$$p(\boldsymbol{r}_t^{B(1:K)} | \boldsymbol{x}_t, r_t^{\mathrm{o}}) = \prod_{i=1}^K N(r_{i,t}^B; \mu_{i,t} + r_t^{\mathrm{o}}, \sigma_{i,t}^2), \tag{10}$$

where the mean of the Gaussian has been modified by the latent offset value r_t^{o}.

The number of visible beacons can vary across time steps and computing the full likelihood can be expensive when many beacons are visible. We can truncate the number of terms in the likelihood product by setting a small value for K and taking a product of the K largest RSS signals. Taking this truncation into account, we can properly estimate the true distribution from this truncated distribution in the following way. Extracting the largest subset $\boldsymbol{r}_t^{B(1:K)}$ from the original RSS vector $\boldsymbol{r}_t^{B(1:M)}$ can be seen that the values of $\boldsymbol{r}_t^{B(1:K)}$ are generated from a truncated distribution with a lower limit. The largest value in the discarded value, $r_{K+1,t}^B$, can be such a lower limit. Let $N_{tr}(r_{i,t}^B; \mu_{i,t} + r_t^{\mathrm{o}}, \sigma_{i,t}^2, r_{K+1,t}^B < r_{i,t}^B)$ be the truncated probability distribution for $r_{i,t}^B$ with a lower limit value $r_{K+1,t}^B$. By approximating this truncated distribution to a normal distribution with the same mean and variance, the observation model for $\boldsymbol{r}_t^{B(1:K)}$ incorporated with the effect of truncation, $q(\boldsymbol{r}_t^{B(1:K)} | \boldsymbol{x}_t, r_t^{\mathrm{o}})$, can be approximated by:

$$q(\boldsymbol{r}_t^{B(1:K)} | \boldsymbol{x}_t, r_t^{\mathrm{o}}) \approx \prod_{i=1}^K N(r_{i,t}^B; \mu_{i,t}^{'} + r_t^{\mathrm{o}}, \sigma_{i,t}^{'2}) \tag{11}$$

where $\mu_{i,t}^{'} + r_t^{\mathrm{o}}$ is the mean and $\sigma_{i,t}^{'2}$ is the variance for the truncated distribution of $r_{i,t}^B$.

The above moments can be obtained as follows [41]:

$$\mu_{i,t}^{'} = \mu_{i,t} + \sigma_{i,t} \frac{\phi(a_{i,t})}{1 - \Phi(a_{i,t})} \tag{12a}$$

$$\sigma_{i,t}^{'2} = \sigma_{i,t}^2 \left[1 + \frac{a_{i,t}\phi(a_{i,t})}{1 - \Phi(a_{i,t})} - \left(\frac{\phi(a_{i,t})}{1 - \Phi(a_{i,t})} \right)^2 \right] \tag{12b}$$

where $a_{i,t} = [r_{K+1,t}^B - (\mu_{i,t} + r_t^{\mathrm{o}})]/\sigma_{i,t}$. $\phi(a)$ and $\Phi(a)$ are the probability distribution function and the cumulative probability function of a standard normal distribution $N(a; 0, 1)$.

By transforming the term on the right in (11), the probability distribution of r_t^{o} conditioned on $\boldsymbol{r}_t^{B(1:K)}$ can be obtained as $p(r_t^{\mathrm{o}} | x_t, \boldsymbol{r}_t^{B(1:K)}) = N(r_t^{\mathrm{o}}; \widehat{r}_t^{\mathrm{o}}, \widehat{\sigma}_t^{\mathrm{o}2})$, where $\widehat{r}_t^{\mathrm{o}}$ is the mean and $\widehat{\sigma}_r^2$ is the variance calculated by:

$$\widehat{r}_t^{\mathrm{o}} = \widehat{\sigma}_t^{\mathrm{o}2} \sum_{i=1}^K \frac{r_{i,t}^B - \mu_{i,t}^{'}}{\sigma_{i,t}^{'2}}, \quad \widehat{\sigma}_t^{\mathrm{o}2} = \left(\sum_{i=1}^K \sigma_{i,t}^{'2} \right)^{-1} \tag{13a}$$

By considering this probability distribution as the observation process of r_t^{o}, and assuming its state transition probability as a normal distribution with the mean r_{t-1}^{o} and the variance $\sigma^{\mathrm{b}2}$, *i.e.* $p(r_t^{\mathrm{o}} | r_{t-1}^{\mathrm{o}}) = N(r_t^{\mathrm{o}}; r_{t-1}^{\mathrm{o}}, \sigma^{\mathrm{b}2})$, the r_t^{o} can be estimated using linear Kalman filter [2]. Let the mean and the variance of the posterior of r_t^{o} be $r_{t|t}^{\mathrm{o}}$ and $\sigma_{t|t}^{\mathrm{o}2}$, respectively. By iteratively updating $r^{\mathrm{o}}_{t|t}$ and $\sigma^{\mathrm{o}2}_{t|t}$ based on the Kalman filter for each particle, the RSS offset can be estimated under location uncertainty. As a result of this extension, the state vector in the particle filter is augmented by additional variables as

$$\boldsymbol{z}_t = (x_t, y_t, f_t, v_t, \theta^{\mathrm{o}}_t, r^{\mathrm{o}}_{t|t}, \sigma^{\mathrm{o}2}_{t|t})^\top. \tag{14}$$

D. Fast Likelihood Computation

The Fast Likelihood Computation (FLC) module decomposes the regression model into a set of many local regression models to speed up the computation of the predicted values of RSS given a location. The number of beacons deployed in very large buildings can reach into hundreds or thousands. In order to build an accurate regression model, the fingerprinting process (*i.e.*, measuring RSS at various points in the building) may result in tens or hundreds of thousands of measurements. This can lead to serious computational issues when the number of fingerprint points reaches this level of magnitude. In the observation model we have described in Section III-C, the nonparametric regression based on a kernel function has a computational complexity time of $\mathcal{O}(N)$ for predicting RSS given a location. The computation grows linearly in the number of fingerprint points N. This is clearly not practical as the particle filter must perform an $\mathcal{O}(N)$ operation, for every observation, for every particle, and for every beacon considered. Therefore, it is necessary to reduce the computational complexity for RSS prediction for large areas. To mitigate this computational complexity issue, we split the regression model into small parts – local models – in input (location) space. For each local model, the computational complexity becomes $\mathcal{O}(N_{\mathrm{sp}})$, where N_{sp} is the average number of training data assigned to each local model by splitting. N_{sp} is much smaller than the total number of fingerprint points N. At test time, we find the top M_{sp} local models closest to the current location estimate $\boldsymbol{x}_t$ and use their kernel-weighted average to compute the predicted values of RSS similar to [42]. The computational complexity to predict value of RSS can be reduced from $\mathcal{O}(N)$ to $\mathcal{O}(N_{\mathrm{sp}}M_{\mathrm{sp}})$. The M_{sp} is typically selected as a small natural number, *e.g.*, $M_{\mathrm{sp}} = 3$, to reduce the computational complexity as much as possible. In our implementation, the input space is split by the k-means clustering algorithm in the training phase.

V. Performance Evaluation

We evaluated our localization system in a real world environment: a shopping mall spanning three multi-story buildings and an underground public space that connects them. We first describe the experimental settings: experimental environment and data collection. Subsequently, we evaluate the overall impact of the proposed improvements on the localization accuracy of the system. We then perform ablative evaluations of the proposed improvements: for each improvement, we compare the localization accuracy of the complete system against the version of the localization approach with the selected improvement removed.

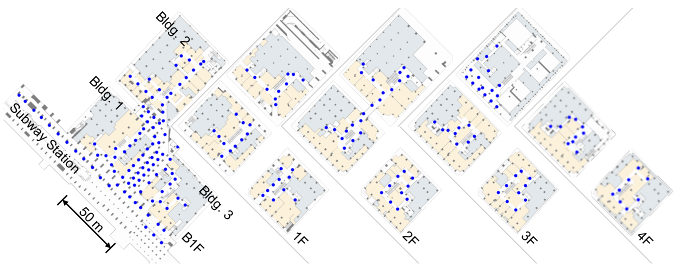


Fig. 2: Floor maps of the experimental site from the basement to the 4th floor. Blue dots show beacon locations.

A. Experimental Settings

1) Environment: Figure 2 displays the floor maps for the experimental site. The testing environment covers three buildings: 1) a building with four floors and one basement, 2) a building with three floors and one basement, and 3) a building with four floors and one basement. The basement floors of these buildings are connected through an underground public pedestrian walkway. The total area of the experimental site is about 21,000 m^2. We extracted the information on accessible areas and floor transition areas (escalators and elevators) from the floor plans. A total of 218 beacons were installed in this environment, with about $5 \sim 10$ m distance between beacons, to enable indoor localization.

2) Data Collection: To evaluate our localization system, we collected fingerprint data and test data using the data acquisition equipment described in the following. Note that we collected both fingerprint and test data during business hours in which the shopping facilities were open. Thus, the evaluation conditions reflected the real world use case since the data were affected by crowds traversing the testing environment.

Data Collection Equipment: To reliably evaluate the localization accuracy, especially in situations where the user is moving, it is important to reduce human error in assigning ground truth locations to fingerprints and test data. Manually labelling all collected data with actual location information is a long and error-prone process. Therefore, we automated the data collection and ground truth assignment procedures using dedicated data collection equipment. Figure 3 shows our equipment, which is composed of the following items:

- a Velodyne VLP-16 LIDAR to record a point cloud of the surrounding environment.
- an Xsens Mti-30 Inertia Measurement Unit (IMU) to compensate for rotational movement of LIDAR during data collection
- an Apple iPhone 6 smartphone used to collect embedded sensor data and Bluetooth RSS data.
- an Apple iPhone 7 smartphone used to collect embedded sensor data and Bluetooth RSS data for evaluation of adaptive signal calibration.
- a laptop computer cabled to other components to simultaneously record LIDAR, IMU and smartphone data.

The collected data were later processed to reconstruct the fingerprint positions with a three-dimensional SLAM algo-

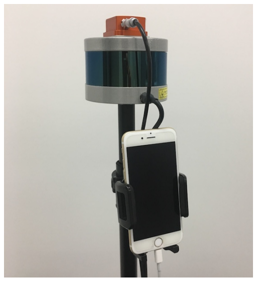


Fig. 3: Data acquisition equipment

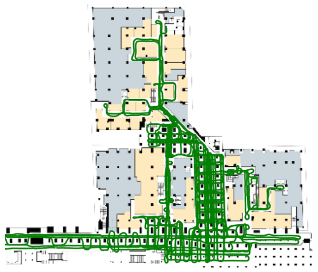
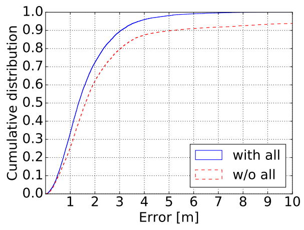
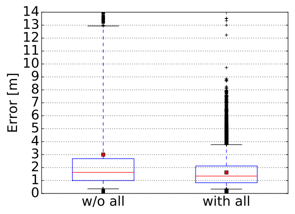


Fig. 4: Basement floor fingerprint collection sequences



(a) Cumulative error distribution



(b) Localization error box plot

Fig. 5: Overall localization error

rithm based on point cloud registration using Normal Distribution Transformation (NDT) [43], [44]. We then tested and validated the collected data by confirming that the projection of the registered point cloud onto a ground plane was in good agreement with the floor plans of the environment.

Fingerprint Collection: We collected fingerprints at about one meter intervals throughout the environment. The data acquisition equipment was fixed to an electric wheelchair (WHILL Model A[1]) during fingerprint collection to keep the movement of the equipment stable, while one experimenter controlled the movement of the wheelchair. We collected one sequence for each floor, per building, and three sequences for the underground public area, which totaled 17 sequences and 17,745 data samples.

Test Data Collection: Differently from the fingerprint data, which only included pairs of locations and respective RSS vectors, our test data also included other sensor data used for the localization. We collected a time-series of iPhone sensor data (accelerometer, attitude, barometric pressure, heading, and BLE RSS data) with correct location labels. Even though the use of SLAM made it easier to assign correct location labels to test data, it lacks semantic information which is important to evaluate performance, such as the time a user reached a target floor using an escalator or an elevator. This additional information was input externally using a smartphone. For the testing, we collected three distinct datasets, defined as follows:

1) Static: On a single floor, walk along one path. Stop every 4 - 10 m for about fifteen seconds. (11,028 seconds, with iPhone 6)
2) Walk: On a single floor, walk along one path, from the starting point to the end point. (8,274 seconds, with iPhone 6)
3) Walk with Floor Transition: Walk along one path, from the starting point to the end point. A floor transition is present using a vertical transportation device, *i.e.*, escalator or elevator. (2,477 seconds, with iPhone 6 and iPhone 7)

To evaluate the performance of the localization with different smartphones, the "Walk with Floor Transition" dataset 3) has been collected with two different devices at the same time. We highlight that the second device (iPhone 7) has a different signal receiving characteristics from the first device (iPhone 6), which is also the fingerprint collection device. On the devices, the update frequencies of accelerometer, attitude, barometric pressure, and heading data are about 100 Hz, 100 Hz, 1 Hz, and 50 Hz, respectively. The report interval of BLE RSS readings is 1 second, and localization errors are calculated for each RSS update.

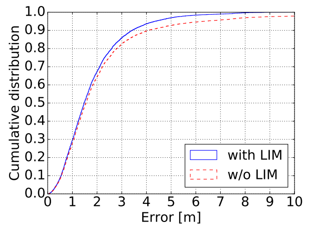
B. Overall Localization Error

We measured the overall localization accuracy on a global dataset, comprising data from the three datasets collected with iPhone 6: 1) "Static", 2) "Walk", and 3) "Walk with Floor Transition". The comparison of localization errors between the base localization algorithm without any improvements and our localization system with all improvements is shown in Figure 5. Figure 5a shows, on the horizontal axis, the localization error, and on the vertical axis, the cumulative distribution of the error. The localization error is calculated as Euclidean distance in (x,y) space. In Figure 5b, 5-percentile, 25-percentile, median, 75-percentile and 95-percentile error is indicated. We also mark the mean as a red point. Although the median error of both the base algorithm (1.6 m) and our system (1.3 m) shows a small difference, in large error cases, the base algorithm without the improvements performs significantly worse than our proposed system. In particular, the 95-percentile error decreased from 12.9 m to 3.8 m, resulting in a mean accuracy improvement from 3.0 m to 1.6 m. These values indicate that our system can manage localization much better in critical cases, which makes it a better approach for real world applications. In the following sections, we assess the impact of the four enhancements to localization: (1) Localization Integrity Monitoring, (2) Floor Transition Control, (3) Adaptive Signal Calibration, and (4) Fast Likelihood Computation.

C. Localization Integrity Monitoring

The effect of the localization integrity monitoring module is evident in situations in which a large number of particles in the particle filter fail to approximate the probability distribution of the localization state. This may happen when the user's movement model diverges from the expected and the resulting errors accumulate. To assess the effect of the localization

[1]http://whill.us/model-a-personal-mobility-device-personal-ev


(a) Cumulative localization error

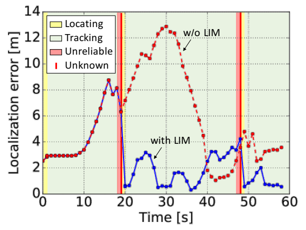
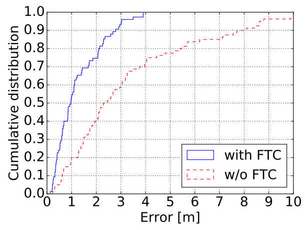

(b) Example time series of localization error

Fig. 6: Effect of Localization Integrity Monitoring


(a) Cumulative localization error during 5 seconds after floor transition

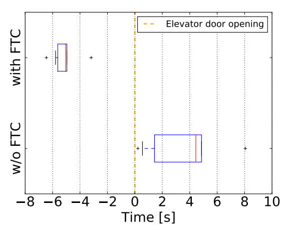

(b) Time of detection of arrival to the floor with respect to the actual arrival

Fig. 7: Effect of Floor Transition Control

integrity monitoring module in this case, we evaluated the localization error with and without the module using the test dataset 2) “Walk”. Figure 6 plots the effects of the localization integrity monitoring module, where Figure 6a is the cumulative distribution of localization error and Figure 6b is an example time series of localization error. In Figure 6a it can be observed that the localization error is reduced by applying the localization integrity monitoring module. To investigate the effect of the localization integrity monitoring module, one example time series that includes catastrophic failures in localization is extracted in Figure 6b. The localization error both with and without localization integrity monitoring is the same until around 20s after localization starts, then the localization error without localization integrity monitoring hits a peak and gradually decreases to the initial level. Instead, the localization integrity monitoring module manages to re-initiate the localization before the error peak, and thus provides a higher localization accuracy.

D. Floor Transition Control

We compare (i) the localization system without the floor transition control module and (ii) the system complemented with our floor transition control module (Section IV-B). We compute the localization error on the test dataset 3) “Walk with Floor Transition” with iPhone 6, which contains routes that include the use of escalators or elevators. The goal of the floor transition control module is to a) relocalize the user correctly upon leaving the elevator and b) notify the user of the floor change in time (about 5 seconds before actually reaching the floor). Because the floor transition control impacts the localization only around floor transitions, we evaluated the errors in a short period (5 seconds) after floor transition is finished.

Figure 7 shows the effect of the floor transition control on the localization: Figure 7a shows the cumulative distribution of the localization error for a short period of time (5 seconds) after floor transition and 7b indicates times when the localization system detected the arrival to the target floor with respect to the actual arrival. We note that the arrival time of the elevator to the target floor was annotated at the moment when the elevator doors started to open. A negative value means that the localization on the floor happened before the actual arrival to the target floor, while a positive value means that the device has been detected on the target floor after the actual arrival. Considering the voice navigation assistant use case, it is preferred that the application notices the arrival to the target floor a few seconds before the actual arrival because it takes few seconds to notify a user of the arrival and next actions to perform by voice instructions.

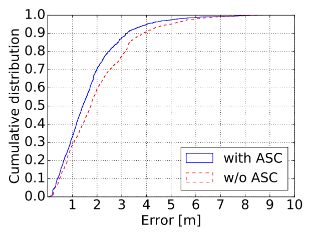
In Figure 7a, the localization error with the floor transition control is visibly better than without floor transition control. This is further backed up by Figure 7b, which indicates that the floor transition control module detected that the device had reached the target floor on average 5 seconds before the actual arrival, which is the desired result. On the other hand, without floor transition control, the average moment in which the floor transition is detected is about 1-5 seconds after reaching the target floor (with about 8 seconds delay in some cases).

E. Adaptive Signal Calibration

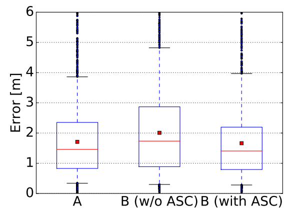
We evaluate the localization with and without the adaptive signal calibration on the test dataset 3) “Walk with Floor Transition” that was recorded by two different devices. For the sake of simplicity, we denote the device for fingerprint collection (Apple iPhone 6) as “A” and the other device (Apple iPhone 7) as “B”. Figure 8 plots the effect of the adaptive signal calibration on localization error, where Figure 8a is the cumulative error distribution and Figure 8b is a boxplot to compare statistical values between device A, device B without adjustment and device B with adjustment. Compared to the previous two modules that impact significantly a limited part of data, this module moderately reduces localization error across the whole sequence, as seen in Figure 8a. To investigate the effect in detail, we compare the error obtained by device B to device A in Figure 8b. It is confirmed that this module reduces the error of device B to the level of error obtained by device A which has no RSS offset.

F. Fast Likelihood Computation

We evaluate the impact of the fast likelihood computation on computation times and localization error. We run the localization system with and without this enhancement on an Apple iPhone 6 smartphone. Figure 9 plots the effect of the fast likelihood computation module, where Figure 9a indicates the time required to process one second of input, and Figure

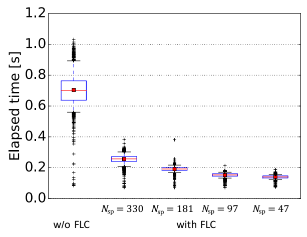


(a) Cumulative localization error by device B

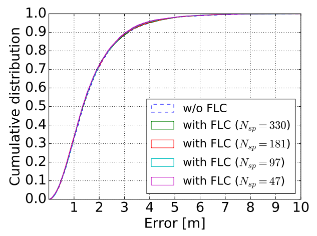


(b) Comparison between device A and device B

Fig. 8: Effect of Adaptive Signal Calibration



(a) Elapsed time to process one second of input



(b) Cumulative localization error

Fig. 9: Effect of Fast Likelihood Computation

9b indicates the cumulative localization error. We investigated the input space partitioning with four different settings: $N_{\text{sp}} = 330, 181, 97, 47$. The top $M_{\text{sp}} = 3$ local models are used to predict RSS values. The other parameters that affect computational times were set to the fixed values ($L = 300$ and $K = 10$). The average elapsed time decreased by 80 percent (from 0.70s to 0.14s) thanks to the the computational efficiency improvement (Figure 9a). At the same time, the localization error does not show any increase (Figure 9b). With this improvement, the localization system achieved a sufficiently small computational burden to run flawlessly on a commodity smartphone device.

VI. Evaluation with Users

We integrate our localization system in a turn-by-turn navigation application [34] and then evaluate the localization accuracy while actual users with visual impairments are traversing the experiment field with the navigation application.

A. Method

We recruited ten participants (4m/6f) with visual impairments (6 legally blind and 4 low vision), with ages ranging from 33 to 54 (M=44, SD=5.9) years old. One participant brought her guide dog while the others used white canes while navigating with the system. During the study, participants were asked to navigate three fixed routes in the shopping mall: 1) starting from the area in front of a subway station on the basement floor to a movie theater on the 3rd floor (177m), 2) starting from the movie theater to a candy shop on the 1st floor (54m), and 3) starting from the shop back to the subway station. The total number of turns in these routes was 26. Note that each route included a transition between floors via an elevator. All participants were asked to wear a waist bag with the phone attached to it in order to free their hands from holding the phone during navigation, which is important for users with visual impairments whose hand is often occupied holding a cane or guide dog's leash. An experimenter followed and videotaped all participants with a camera close behind them. We visually annotated participants' actual location every second, except for when they were inside an elevator. We also annotated their turn performance, *i.e.*, whether they successfully made a turn without the experimenter's help.

B. Results

We evaluate the localization error between users' actual locations and corresponding estimated locations. The number of annotated location points is 7641 from the all routes for all the participants. We obtained 1.7m mean localization error and 3.2 m localization error at 95-percentile during navigation. We also investigate the performance of navigation relying on our localization system. Of the 260 turns in total (26 turns per participant), 243 turns (93.5%) were successful without the experimenter's help. The results demonstrate the proposed system can mostly provide actual visually impaired users with location sufficient for navigation.

VII. Conclusions

To provide automated turn-by-turn indoor navigation assistance for individuals with visual impairments, it is essential to develop a localization system that can achieve high levels of accuracy in building-scale real world environments. In this paper, we considered the challenges which must be overcome to enable navigational assistance in a concrete real-world multi-story scenario. To address the challenges within a unified framework, we designed and implemented a localization system that enhances a probabilistic localization algorithm with a series of innovations in order to achieve accurate real-time localization. We performed a series of experiments with ground truth data collected in a large indoor environment composed of three multi-story buildings and an underground passageway. The experimental evaluations validated the effect of the enhancement modules to improve the localization accuracy and provide real-time navigation capabilities in real-world scenarios. Specifically, the mean localization error decreased from 3.0m to 1.6m. We also evaluated the practical performance of the system in a study with visually impaired participants, and the results demonstrate that the proposed localization system helps their independent mobility.

References

[1] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *Proc. of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 147–158.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[3] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, “A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned,” in *Proc. of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 178–189.

[4] A. W. Tsui, Y.-H. Chuang, and H.-H. Chu, “Unsupervised learning for solving RSS hardware variance problem in WiFi localization,” *Mobile Networks and Applications*, vol. 14, no. 5, pp. 677–691, 2009.

[5] L. Li, G. Shen, C. Zhao, T. Moscibroda, J.-H. Lin, and F. Zhao, “Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service,” in *Proc. of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, 2014, pp. 459–470.

[6] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu, “Enhancing WiFi-based localization with visual clues,” in *Proc. of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015, pp. 963–974.

[7] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *Proc. IEEE INFOCOM*, vol. 2. IEEE, 2000, pp. 775–784.

[8] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, 2007.

[9] R. Harle, “A survey of indoor inertial positioning systems for pedestrians,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, 2013.

[10] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, “Mobility increases localizability: A survey on wireless indoor localization using inertial sensors,” *ACM Computing Surveys*, vol. 47, no. 3, pp. 54:1–54:34, 2015.

[11] S. He and S.-H. G. Chan, “Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 2016.

[12] B. Ferris, D. Haehnel, and D. Fox, “Gaussian processes for signal strength-based location estimation,” in *Proc. of Robotics Science and Systems*, 2006.

[13] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, “Indoor positioning in Bluetooth networks using fingerprinting and lateration approach,” in *Proc. of International Conference on Information Science and Applications*. IEEE, 2011.

[14] R. Faragher and R. Harle, “Location fingerprinting with Bluetooth Low Energy beacons,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, 2015.

[15] J. Wang and D. Katabi, “Dude, where’s my card?: RFID positioning that works with multipath and non-line of sight,” *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 51–62, 2013.

[16] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, 2005.

[17] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, “ALPS: A Bluetooth and ultrasound platform for mapping and localization,” in *Proc. of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 73–84.

[18] Z. Yang, Z. Zhou, and Y. Liu, “From RSSI to CSI: Indoor localization via channel response,” *ACM Computing Surveys*, vol. 46, no. 2, pp. 25:1–25:32, 2013.

[19] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-level localization with a single WiFi access point.” in *Proc. of NSDI*, 2016, pp. 165–178.

[20] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, “A probabilistic approach to wlan user location estimation,” *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.

[21] Y. Gwon and R. Jain, “Error characteristics and calibration-free techniques for wireless LAN-based location estimation,” in *Proc. of the Second International Workshop on Mobility Management & Wireless Access Protocols*. ACM, 2004, pp. 2–9.

[22] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, “Indoor localization without the pain,” in *Proc. of the 16th Annual International Conference on Mobile Computing and Networking*. ACM, 2010, pp. 173–184.

[23] Z. Yang, C. Wu, and Y. Liu, “Locating in fingerprint space: wireless indoor localization with little human intervention,” in *Proc. of the 18th Annual International Conference on Mobile Computing and Networking*. ACM, 2012, pp. 269–280.

[24] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, “Does BTLE measure up against WiFi? a comparison of indoor location performance,” in *Proc. of the 20th European Wireless Conference*, 2014.

[25] F. Palumbo, P. Barsocchi, S. Chessa, and J. C. Augusto, “A stigmergic approach to indoor localization using Bluetooth Low Energy beacons,” in *Proc. of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2015.

[26] O. Woodman and R. Harle, “Pedestrian localisation for indoor environments,” in *Proc. of the 10th International Conference on Ubiquitous Computing*. ACM, 2008, pp. 114–123.

[27] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, “A reliable and accurate indoor localization method using phone inertial sensors,” in *Proc. of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 421–430.

[28] N. A. Giudice and G. E. Legge, “Blind navigation and the role of technology,” *The Engineering Handbook of Smart Technology for Aging, Disability, and Independence*, pp. 479–500, 2008.

[29] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, “Indoor human navigation systems: A survey,” *Interacting with Computers*, vol. 25, no. 1, pp. 21–33, 2013.

[30] J.-E. Kim, M. Bessho, S. Kobayashi, N. Koshizuka, and K. Sakamura, “Navigating visually impaired travelers in a large train station using smartphone and Bluetooth Low Energy,” in *Proc. of the 31st Annual ACM Symposium on Applied Computing*. ACM, 2016, pp. 604–611.

[31] S. A. Cheraghi, V. Namboodiri, and L. Walker, “Guidebeacon: Beacon-based indoor wayfinding for the blind, visually impaired, and disoriented,” in *Proc. of IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2017, pp. 121–130.

[32] D. Ahmetovic, C. Gleason, K. M. Kitani, H. Takagi, and C. Asakawa, “NavCog: turn-by-turn smartphone navigation assistant for people with visual impairments or blindness,” in *Proc. of the 13th Web for All Conference*. ACM, 2016, pp. 9:1–9:2.

[33] D. Ahmetovic, C. Gleason, C. Ruan, K. Kitani, H. Takagi, and C. Asakawa, “NavCog: a navigational cognitive assistant for the blind,” in *Proc. of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 2016, pp. 90–99.

[34] D. Sato, U. Oh, K. Naito, H. Takagi, C. Asakawa, and K. Kitani, “NavCog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment,” in *Proc. of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2017, pp. 270–279.

[35] D. Ahmetovic, M. Murata, C. Gleason, E. Brady, H. Takagi, K. Kitani, and C. Asakawa, “Achieving practical and accurate indoor navigation for people with visual impairments,” in *Proc. of the 14th Web for All Conference*, 2017, pp. 31:1–31:10.

[36] A. Brajdic and R. Harle, “Walk detection and step counting on unconstrained smartphones,” in *Proc. of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2013, pp. 225–234.

[37] K. P. Murphy, *Machine Learning. A Probabilistic Perspective*. MIT press, 2012.

[38] J. Fink and V. Kumar, “Online methods for radio signal mapping with mobile robots,” in *Proc. of the 2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 1940–1945.

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[40] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, “Barometric phone sensors: More hype than hope!” in *Proc. of the 15th Workshop on Mobile Computing Systems and Applications*. ACM, 2014, pp. 12:1–12:6.

[41] W. H. Greene, *Econometric Analysis (7th Edition)*. Pearson Education, 2011.

[42] D. Nguyen-Tuong and J. Peters, “Local gaussian process regression for real-time model-based robot control,” in *Proc. of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 380–385.

[43] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An open approach to autonomous vehicles,” *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[44] M. Magnusson, “The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection,” Ph.D. dissertation, Örebro universitet, 2009.